

Flugdynamik, Multisensor-Navigation und Steuerung skalierbarer Out-/Indoor-Multicopter UAV

Reiner Jäger | Jan Zwiener
Hochschule Karlsruhe – Technik und Wirtschaft (HSKA)
Insitut für Angewandte Forschung (IAF)
Moltkestraße 30, 76133 Karlsruhe
reiner.jaeger@hs-karlsruhe.de | jan.zwiener@hs-karlsruhe.de

Abstract

The NAVKA navigation technologies for distributed GNSS/MEMS/MOEMS sensors, developed since 2008 at IAF/HSKA and have been awarded “Bronze” at the European Satellite Navigation Competition (ESNC) 2014. The NAVKA flight-control system (algorithms, software and hardware) is one part of the above technologies. It can be used for the navigation and control of UAV and manned multicopters. Hereby any n-propeller multicopter can be designed scalable in respect to the 3D propeller layout and orientation in the body (b) frame, the applications, the size and the payload, as well as the further system equipment in case of UAV.

The NAVKA flight navigation and control system is based on at least one, generally a redundant number of m GNSS/MEMS and camera (MOEMS) multisensor platforms (p) situated on the body (b).

At first the NAVKA multisensor-multiplatform concept for the estimation of the navigation state vector $\mathbf{y}(t)$ of distributed GNSS/MEMS/MOEMS sensor data is presented.

A second focus is then put on the flight physics and the physical and mathematical background of the control of an n-propeller multicopter. The general relations between the above 15-parameter navigation state vector $\mathbf{y}(t)$ and its changes and the n-dimensional control state vector $\mathbf{u}(t)$, given by the n propeller rotation rates, is derived.

Based on the above general physical background and equations of the flight-dynamics of an n-propeller body, the estimation of the flight control vector $\mathbf{u}(t)$ and its relation to a PID-controller are presented. At first the navigation state $\mathbf{y}(t)$ of one, or a number of m distributed navigation boxes, are regarded as one control unit for the estimation of $\mathbf{u}(t)$. The respective NAVKA PID-controller concept is derived, and the developed flight-control system (hard- and software) NAVKArine-FC4 is presented.

In order to receive a secure flight control system, the redundancy of the above mentioned m physically separated navigation boxes and their navigation state $\mathbf{y}(t)_i$ and control state estimates $\mathbf{u}(t)_i$, respectively can be used to provide in real-time a final robust M-estimate of $\mathbf{u}(t)$. Like for the estimation of the navigation state estimation $\mathbf{y}(t)$ from sensor data, a robust L_1 -norm (instead of least squares) leads to a secure control state estimate $\mathbf{u}(t)$ in case of eventual errors in the single estimation $\mathbf{u}(t)_i$ used as input. Based on SIMPLEX-algorithms (instead of iterative least squares), the L_1 -norm further allows to consider different kind of restrictions, e. g. in terms of in-equations on the control state vector $\mathbf{u}(t)$ of the propeller rotation rates.

1 Einleitung

Die weitreichenden und nachhaltigen Potenziale der NAKVA FuE ([1], [2]) basieren hinsichtlich der Schätzung des Navigationszustandsvektors $\mathbf{y}(t)$

$$\begin{array}{l} \text{Navigations-} \\ \text{Zustandsvektor} \end{array} \quad \mathbf{y}(t) = \begin{bmatrix} x^e & y^e & z^e & \dot{x}^e & \dot{y}^e & \dot{z}^e & r^e & p^e & y^e \\ \ddot{x}^e & \ddot{y}^e & \ddot{z}^e & \omega_{eb,x}^b & \omega_{eb,y}^b & \omega_{eb,z}^b & \mathbf{s} \end{bmatrix}^T \quad (1a)$$

zum einen auf verschiedenen

- Allgemeinen Vorhersagemodellen $\mathbf{y}(t)_{t-\Delta t}, \mathbf{C}_{\mathbf{y}(t), t-\Delta t}$ (2a) zum Navigationszustandsvektor $\mathbf{y}(t)$ ([5], [15], [28])

und zum anderen aus den GNSS/MEMS/MOEMS-Sensordaten $\mathbf{l}=\mathbf{l}(\mathbf{y}(t))$ (2b). Der Navigationszustandsvektor $\mathbf{y}(t)$ (1a) umfasst dabei in obiger Reihenfolge die Parameter der 3D-Position, -Geschwindigkeit, -Orientierung, -Beschleunigung und Drehraten des sich in allgemeiner Bewegung, im Sonderfall in Ruhe, befindlichen zu navigierenden Objekts (dem Body (b)).

Die Navigationskomponente $\mathbf{y}(t)$ (1a) und die entsprechende GNSS/MEMS/MOEMS Hardware (Abb. 2) hat im Regelkreis der Flugsteuerung (Kap. 3 bis 6)

zur permanenten Herbeiführung des Soll-Zustandsvektors $\mathbf{y}(t)_{\text{Soll}}$ die Aufgabe, fortlaufend den Ist-Zustandsvektor $\mathbf{y}(t)_{\text{Ist}}$ des Bodys mittels der auf dem Body (b) verorteten multisensorischen Navigationsplattformen (Abb. 2) oder Einzelsensoren (Abb. 1) bereitzustellen. Zur physikalisch begründeten Zustandsvorhersage $\mathbf{y}(t)_{t-\Delta t}$ (2a) tritt als zweite Komponente im NAVKA-Konzept die algorithmische Modellierung verteilter GNSS/MEMS/MOEMS Sensoren hinzu. Diese erfolgt in der Regel als

- Tiefe Kopplung der Rohdaten $\mathbf{I}=\mathbf{I}(\mathbf{y}(t))$ (2b) der GNSS-Sensoren (Code-, Phasen- und Dopplermessungen), der MEMS-Sensoren (Beschleunigungsmesser, Gyroskope, Magnetometer, Barometer) sowie der MOEMS bzw. Kameraoptik (Marker, Virtuelle Marker, visuelle Odometrie (VO)).

Für GNSS und weitere Positions- und Geschwindigkeitsinformation liefernde Komponenten (z. B. WiFi, Odometer) ist auch eine enge Kopplung („Tight Coupling“) mit den übrigen Sensordaten möglich. Die mathematische Modellbildung der Sensorkomponenten erfolgt bei NAVKA ([1], [2]) dabei in einem

- Multisensor-Multiplattform-Design (Abb. 1, Kap. 2, [4]).

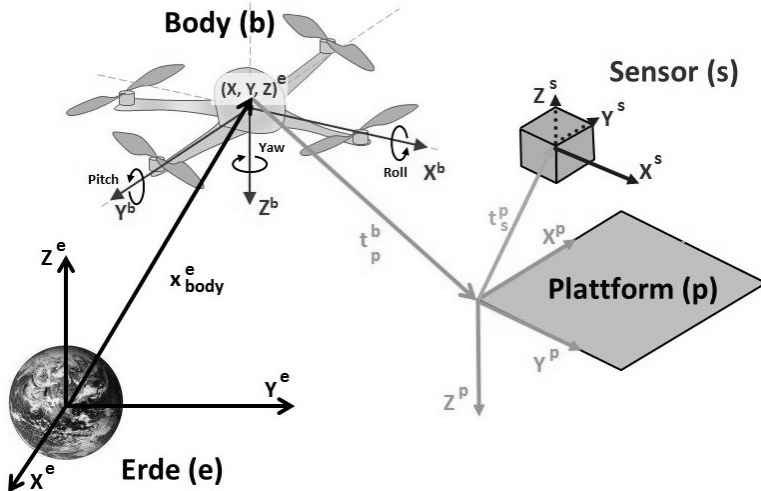
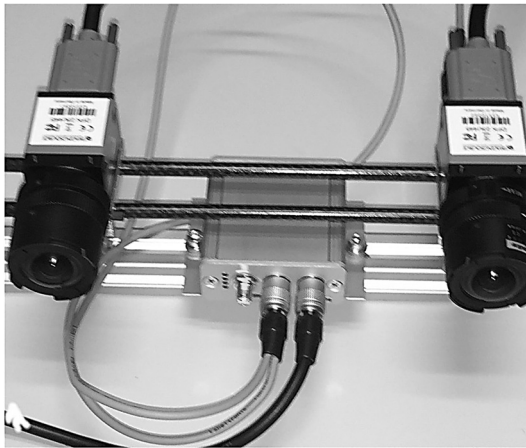


Abb. 1: Navigation eines UAV als Body (b) mit verteilten Sensoren am Beispiel eines Quadropters ($n=4$ Propeller Multicopter) und einer Plattform ($m=1$)

Bei diesem weiteren Alleinstellungsmerkmal erhalten die GNSS-, MEMS- und MOEMS- bzw. Kamera-Einzelsensoren s_{ij} zum einen den Plattformindex (i), unter welchem die Sensorposition und -Orientierung auf der j-ten Plattform p_j verwaltet wird. Der Index j der j-ten Plattform p_j – im Sonderfall auch ein Smartphone oder Tablet – verwaltet die Position und Orientierung von im Objektkoordinatensystem, dem Body-System (b) (Abb. 1). Erst dieses algorithmische Design (NAVKA „Leverarm-Konzept“) ermöglicht die Schätzung des 15-parametrischen Navigationszustandsvektors y (1a) aus einer beliebigen Anzahl auf dem zu navigierenden Body (b) verteilter Plattformen (p) und Sensoren (s).



*Abb. 2: NAVKArine-GIMC-Navigationsbox bzw. -Plattform.
In der Mitte die GNSS/MEMS Komponente (L1-GNSS, MEMS-Accelerometer/Gyroskop-Triade, MEMS-Magnetometer und -Barometer.
Rechts/links die Stereokamera-Komponente*

Die „Verortung“ der verteilten Sensoren s_{ij} umfasst dabei bis zu 11 sog. Leverarm-Parameter. Der – z. B. im Fall Fahrzeugnavigation – auftretende Sonderfall Plattform p_j gleich Body (b) ist dabei mit eingeschlossen. Die dritte Komponente und die weiteren NAVKA Alleinstellungsmerkmale beziehen sich auf folgende Eigenschaften zur Navigationszustandsschätzung $y(t)$ (1a):

- Integration von Zusatzinformationen zum geometrischen und physikalischen Bewegungsprofil ([28]),
- Einbeziehung von Zusatzsensoren/-Informationen zum Raumbezug z. B. Gleichungen/Ungleichungen [28] über eine zugleich auch robuste L_1 -Norm Schätzung mittels SIMPLEX-Algorithmen (Kap. 5, 6 [28]),

- Einbeziehung von Sensorkalibrierungsparametern s und -Algorithmen,
- Robuste Schätzung (z. B. L_1 -Norm [7]) des Zustandsvektors $\mathbf{y}(t)$ und auftretender stochastischer Modellkomponenten.

Der Stellgrößenvektor $\mathbf{u}(t)$ einer UAV-Flugsteuerung umfasst als n -dimensionale Stellgröße des Steuerungsregelkreises (Kap. 3 bis 6) eines Multicopters mit

$$\begin{aligned} \text{Stellgrößen-} & \mathbf{u}(t) = [\omega_1, \dots, \omega_i, \dots, \omega_n]^T \text{ bzw.} \\ \text{vektor} & \mathbf{u}(t) = [\Delta\omega_1, \dots, \Delta\omega_i, \dots, \Delta\omega_n]^T \end{aligned} \quad (1b)$$

die Drehzahlen ω_i der n Propeller bzw. deren fortlaufende Änderungen $\Delta\omega_i$. Die zugehörige Regeldifferenz, auch Regelabweichung, $\mathbf{e}(t)$ (siehe (6)) ergibt sich als fortlaufende Korrektur aus dem über die Kanäle des Pilotenjoysticks, der UAV-Fernsteuerung oder dem Autopiloten angeforderten Navigationsoll $\mathbf{y}(t)_{\text{Soll}}$ gegenüber dem GNSS/MEMS/MOEMS-sensorisch in der Navigationskomponente $\mathbf{y}(t)$ (1a) der Flugsteuerung ermittelten Istzustand $\mathbf{y}(t)_{\text{Ist}}$.

Aus physikalischen Gründen wird $\mathbf{u}(t)$ zweistufig berechnet (Kap. 3 bis 4). Der als PD bzw. PID-Regler realisierbaren Stufe 1 liegen die Bewegungsgleichungen des Body (b) in Funktion von Gesamtschubkraft und -Drehmoment (5a,b,c) zugrunde. In Stufe 2 werden diese in Stufe 1 aktualisierten Größen in neue Propellerdrehzahlen ω_i bzw. den Änderungen $\Delta\omega_i$ überführt.

2 NAVKA-Multisensor-Multiplattform-Navigationskonzept

Die fortlaufende Ermittlung des Navigationszustandsvektors $\mathbf{y}(t)$ (1a) zum Zeitpunkt t erfolgt in der Fahrzeug- und Flugnavigation i. A. als gemeinsame Parameterschätzung im nachfolgenden Gauß-Markov-Modell (2a,b) bestehend aus der Komponente der Zustandsvorhersage $\mathbf{y}(t)_{t-\Delta t}$ (2a) und der zweiten Komponente der Sensordaten $\mathbf{I}(t)$ (2b) samt den zugehörigen stochastischen Modellen:

$$\begin{aligned} \text{Gauß-Markov-Modell} & \mathbf{y}(t)_{t-\Delta t}, \mathbf{C}_{\mathbf{y}(t), t-\Delta t} \text{ und } \mathbf{I}(t) = \mathbf{I}(\mathbf{y}(t)), \mathbf{C}_{\mathbf{I}(t)} \\ \text{zur Schätzung von } \mathbf{y}(t) & \end{aligned} \quad (2a,b)$$

Was unterschiedliche Vorhersagemodelle zur Berechnung von $\mathbf{y}(t)_{t-\Delta t}$, $\mathbf{C}_{\mathbf{y}(t), t-\Delta t}$ (2a) angeht, so wird auf [5] und [15] verwiesen.

Erfolgt die Navigationszustandsschätzung $\mathbf{y}(t)$ (1a) im Gauß-Markov-Modell (2a,b) als Kleinste-Quadrate-Schätzung (L_2 -Norm) [7], so kann sie in Äquivalenzumformung auch in die Notation einer klassischen Kalman-Filterung überführt werden [14]. Auch eine auf (2a,b) basierte robuste M-Schätzung (z. B. L_1 -Norm) [7] kann wiederum in eine entsprechende iterative robuste Kalman-Filterung überführt werden [14]. Die Grenzen der klassischen Kleinste-Quadrate- und M-Schätzung im Gauß-Markov-Modell (2a,b) und der als Äquivalenzmodell damit einhergehenden Kalman-Filterungen sind aber – gerade auch bei hochfrequenten Echtzeitanwendungen – dann erreicht, wenn – wie in der Navigation und Steuerung häufig der Fall (Kap. 3 bis 6) – weitere Information zum Zustandsparameterraum $\mathbf{y}(t)$ (1a) bzw. zur Bestimmung der Regelgröße $\mathbf{u}(t)$ (1b) (z. B. Ungleichungen) einfließen soll. Hierbei erweist sich die auf SIMPLEX-Algorithmen ([12], [13]) basierte und zugleich auch Modellfehler-robuste L_1 -Norm als ebenso hinreichende und wie potenziell innovative Methode. Dies wird in den Kap. 5 und 6 anhand der robusten Schätzung des Stellgrößenvektors $\mathbf{x}(t)=\mathbf{u}(t)$ aus den Einzellösungen $\mathbf{u}(t)_i$ redundanter Navigations- und Steuerungsboxen gezeigt.

Was die Modellierung der Sensorbeobachtungskomponente $\mathbf{l}(t)$ (2b) bei der Navigationszustandsschätzung $\mathbf{y}(t)$ anbelangt, so zeichnet sich das NAVKA-Konzept hier dadurch aus, dass die GNSS-, MEMS- und MOEMS- bzw. Kamera-Sensoren (Abb. 1, Abb. 2) als Einzelsensoren in den NAVKA-Algorithmen und Datenstrukturen s_{ij} zweifach indiziert sind. Der Index i verwaltet den Sensortyp und dessen Position und -Orientierung auf der j -ten Plattform. Der Index j verwaltet die Verortung der Plattform p_j im Koordinatensystem des zu navigierenden Objekts, dem Body (b) System (Abb. 1). Dieses algorithmische Design („NAVKA Leverarm-Konzept“) ermöglicht eine allgemeine, redundante, selbstkalibrierende und optimale Schätzung des im allgemeinen 15-parametrigen Navigationszustandsvektors $\mathbf{y}(t)$ (1a) aus einer beliebigen Anzahl auf dem zu navigierenden Body (b) verteilter GNSS- und MEMS-Plattformen bzw. -Sensoren s_{ij} . Algorithmisch sind dabei insgesamt elf Parameter zur Verortung von Sensoren s_{ij} und Plattformen p_j auf dem Body (b) mitzuführen (Abb. 1). Dies sind erstens die drei Parameter des Translationsvektors $\mathbf{t}_{p_j}^b$ der j -ten Plattform p_j im Body-Frame (b), zweitens die drei Parameter der Rotationsmatrix $\mathbf{R}_{p_j}^b$ der j -ten Plattform p_j gegenüber dem Body-Frame (b), drittens die drei Translationsparameter $\mathbf{t}_{s_{ij}}^{p_j}$ des i -ten Sensors s_{ij} im Koordinatensystem der j -ten Plattform p_j und viertens die zwei Parameter $(\alpha, \delta)_{s_{ij}}$ der Sensororientierung auf der Plattform p_j .

Die damit mögliche tiefe Kopplung soll anhand eines i-ten auf der j-ten Navigationsplattform platzierten Beschleunigungssensors s_{ij} aufgezeigt werden. Dessen am Sensorort gemessene eindimensionale Beschleunigung $a_{s_{ij}}$ geht mit dem durch $(\alpha, \delta)_{s_{ij}}$ parametrisierten Sensororientierungsvektor $\mathbf{r}_{s_{ij}}^{p_j}$ aus dem am Plattformrahmen p_j anliegenden Beschleunigungsvektor $\mathbf{a}_{s_{ij}}^{p_j}$ nach (3a1,2) hervor. Mit (3a3) wird der anliegende Beschleunigungsvektor $\mathbf{a}_{s_{ij}}^e$ vom erdfesten System (e) ins Plattformsystem (p) $\mathbf{a}_{s_{ij}}^{p_j}$ (3a1) überführt. Mit (3a3) wird der anliegende Beschleunigungsvektor $\mathbf{a}_{s_{ij}}^e$ vom erdfesten System (e) ins Plattformsystem (p) $\mathbf{a}_{s_{ij}}^{p_j}$ (3a1) überführt.

$$\text{Beschleunigungsmessung} \quad \mathbf{a}_{s_{ij}} = \mathbf{r}_{s_{ij}}^{p_j T} \cdot \mathbf{a}_{s_{ij}}^{p_j} \quad (3a1)$$

$$l(t) =: \mathbf{a}_{s_{ij}}^e \quad \mathbf{r}_{s_{ij}}^{p_j T} = (\cos \delta_{ij} \cdot \sin \alpha_{ij}, \cos \delta_{ij} \cdot \cos \alpha_{ij}, \sin \delta_{ij}) \quad (3a2)$$

$$\text{– Teil 1 –} \quad \mathbf{a}_{s_{ij}}^{p_j} = (\mathbf{R}_{p_j}^b)^T \cdot \mathbf{R}_e^b(r, p, y) \cdot \mathbf{a}_{s_{ij}}^e \quad (3a3)$$

Für den aus der Gesamtbeschleunigung $\ddot{\mathbf{x}}(t)_{s_{ij}}^e$ des Sensors, der Gravitations-, der Coriolis- und der Zentrifugalbeschleunigung – jeweils am Sensorort und dem bzgl. des Navigationszustandsvektors $\mathbf{y}(t)$ (1) interessierenden erdfesten Bezugsrahmen (e) – zusammengesetzten Beschleunigungsvektor $\mathbf{a}_{s_{ij}}^e$ gilt (3b). Durch Einsetzen von (3b) in (3a3), und (3a3) wiederum in die Beobachtungsgleichung (3a1), erhalten wir die Parametrisierung der Sensormessgröße, hier der Beschleunigung $a_{s_{ij}}$ am Sensorort, im erdfesten Bezugsrahmen (e) in Gestalt von (3c).

$$\text{Beschleunigungsmessung} \quad \mathbf{a}_{s_{ij}}^e = [\ddot{\mathbf{x}}(t)^e - \mathbf{g}^e(\mathbf{x}) + 2 \cdot \boldsymbol{\Omega}_{ie}^e \cdot \dot{\mathbf{x}}(t)^e + \boldsymbol{\Omega}_{ie}^e \cdot \boldsymbol{\Omega}_{ie}^e \cdot \mathbf{x}(t)^e]_{s_{ij}} \quad (3b)$$

$$l(t) =: \mathbf{a}_{s_{ij}}^e \quad \mathbf{a}_{s_{ij}}^{p_j} = (\mathbf{r}_{s_{ij}}^{p_j})^T \cdot (\mathbf{R}_{p_j}^b)^T \cdot \mathbf{R}_e^b(r, p, y) \cdot [\ddot{\mathbf{x}}(t)^e - \mathbf{g}^e(\mathbf{x}) + 2 \cdot \boldsymbol{\Omega}_{ie}^e \cdot \dot{\mathbf{x}}(t)^e + \boldsymbol{\Omega}_{ie}^e \cdot \boldsymbol{\Omega}_{ie}^e \cdot \mathbf{x}(t)^e]_{s_{ij}} \quad (3c)$$

– Teil 2 –

Die Verknüpfung von (3c) mit dem zu navigierenden Body (b) erfolgt über die o. g. 11-parametrische Ortsverknüpfung von Sensoren s_{ij} , Plattformen p_i und dem damit zu navigierenden Body (b) in Gestalt der „Leverarm“ (Hebelarm) Beziehung (3d):

Allgemeine Sensorverortung
 Multisensor-Multiplattform
 „Leverarm“-Gleichung

$$\mathbf{x}_{s_{ij}}^e = \mathbf{x}_b^e + \mathbf{R}_b^e(r, p, y) \cdot [\mathbf{t}_{p_j}^b + \mathbf{R}_{p_j}^b \cdot \mathbf{t}_{s_{ij}}^{p_j}] \quad (3d)$$

Durch Einsetzen von (3d) und der entsprechenden ersten und zweiten zeitlichen Ableitung in (3c) erhalten wir, entsprechend der Beobachtungskomponente (2b), das finale funktionale Modell der Parametrisierung der Beschleunigungsmessung $\mathbf{l}(t) = \mathbf{a}_{s_{ij}}$ des i -ten Beschleunigungssensors s_{ij} in tiefer Kopplung („Deep Coupling“) der physikalischen Sensorrohdaten an den Navigationszustandsvektor $\mathbf{y}(t)$ (1a) des Bodys (b). In der tiefen Kopplung im NAVKA Multisensor-Multiplattform-Konzept sind die Rohdatenbeobachtungen $\mathbf{a}_{s_{ij}}$ (3c,d) der auf dem Body (b) verteilten Beschleunigungssensoren s_{ij} , so direkt mit dem interessierenden Body-Navigationszustandsvektor $\mathbf{y}(t)$ (1a) verknüpft. Das obige für den Beschleunigungssensor aufgezeigte Multisensor-Multiplattform-Konzept gilt auch für alle weiteren Navigationssensoren (Verbesserungsgleichungen (2b) weiterer Sensoren (siehe [15], [28])). Alle Sensoren tragen somit mit ihren physikalischen Rohdaten direkt zur Zustandsschätzung $\mathbf{y}(t)$ (1a) des Bodys (b) bei. Die damit verbundene Vermeidung von Mehrfachparametrisierungen impliziert eine optimale, robust [7] durchführbare und selbstkalibrierende Zustandsschätzung $\mathbf{y}(t)$ (1a) aus der Vorhersagekomponente $\mathbf{y}(t)_{t-\Delta t}$ (2a) und der Datenkomponente verschiedener verteilter Sensormessungen $\mathbf{l}(t)$ (2b) – wie im obigen Beispiel der Beschleunigungsmessung $\mathbf{l}(t) =: \mathbf{a}_{s_{ij}}$ (3c) – im Gauß-Markov-Modell (2a,b) [7].

Demgegenüber bedeutet die zweitstufige enge Kopplung („Tight Coupling“) eine suboptimale Integration von Sensoren und deren Daten als sog. direkte Beobachtungen der Navigationszustandsparameter $\mathbf{y}(t)$ (1a). Paradebeispiel für „Tight-Coupling“ ist GNSS. Hier können aus den GNSS-Rohdaten (Code-, Phasen- und Dopplermessungen, Abb. 1) mit GNSS-Algorithmen Positions- und Geschwindigkeitsinformationen – als Teilvektoren von $\mathbf{y}(t)$ (1a) – in einer separaten Stufe geschätzt werden. Für eine aus den GNSS-Rohdaten separat berechnete GNSS-Positionsinformation $\mathbf{l}(t) =: \mathbf{x}_{\text{GNSS-}s_{ij}}^e$ erhalten wir im NAVKA-Multisensor-Multibody-Konzept – im Fall von „Tight Coupling“ – dann in Stufe 2

die Beobachtungsgleichung (4a). Im Sonderfall „Plattform“ $p_j = \text{Body (b)}$ geht (3a) mit $\mathbf{t}_{p_j}^b = \mathbf{0}$ und $\mathbf{R}_{p_j}^b = \mathbf{I}$ in (4b) über.

$$\begin{array}{l} \text{GNSS} \\ \text{Positionsbeobachtungen} \end{array} \quad \mathbf{x}_{\text{GNSS}-s_{ij}}^e = \mathbf{x}_b^e + \mathbf{R}_b^e(r, p, y) \cdot [\mathbf{t}_{p_j}^b + \mathbf{R}_{p_j}^b \cdot \mathbf{t}_{s_{ij}}^{p_j}] \quad (4a)$$

$$\mathbf{I}(t) =: \mathbf{x}_{\text{GNSS}-s_{ij}}^e(t) \quad \mathbf{x}_{\text{GNSS}-s_{ij}}^e = \mathbf{x}_b^e + \mathbf{R}_b^e(r, p, y) \cdot \mathbf{t}_{s_{ij}}^b \quad (4b)$$

Die Beziehungen (4a,b) erlauben im „Tight-Coupling“ die Bestimmung von Position $(x,y,z)^e$ und Orientierung $(r,p,y)^e$ eines Body (b) aus den Positionen $\mathbf{x}_{\text{GNSS}-s_{ij}}^e$ einer Anzahl auf dem Body (b) verorteter GNSS-Sensorplattformen bzw. direkt auf dem Body (b) verorteter einzelner GNSS-Sensoren als direkte Beobachtungen $\mathbf{I}(t)$.

Der Nachteil der zweistufigen engen Kopplung („Tight Coupling“) wird im o. g. GNSS-Beispiel (4a,b) aber sofort deutlich: Es ist vorauszusetzen, dass die aus den GNSS-Rohdaten zu gewinnende Positionsbeobachtungen $\mathbf{x}_{\text{GNSS}-s_{ij}}^e$ für die enge Kopplung (4a,b) bereits in Stufe 1 berechenbar sein müssen. D. h., das „Tight-Coupling“ setzt – im Gegensatz zur allgemeineren tiefen Rohdatenkopplung (Beispiel (3c,d)) – in diesem Fall die Rohdaten von mindestens 4 Satelliten in einer Berechnungsstufe 1 voraus.

Die Einführung der Verortung (3d) als Bedingung im GMM (2a,b) – im Fall von GNSS in der Form (4a,b) – ermöglicht wiederum eine tiefe Kopplung („Deep-Coupling“) von GNSS unter direkter Parametrisierung der Rohdaten $\mathbf{x}_{\text{GNSS}-s_{ij}}^e$. Mit dieser Alternative – „Verortung (3d) als Bedingungsgleichung“ – kann hier die explizite Reparametrisierung bzgl. der Bodyposition \mathbf{x}_b^e (b) verorteten GNSS-Positionsbeobachtungen $\mathbf{x}_{\text{GNSS}-s_{ij}}^e$ (4a,b) erfolgen. Damit kann die Entwicklung von GNSS-Rohdatenalgorithmen im NAVKA Multisensor-Multiplattform-Navigationskonzept auf den Sensor- bzw. Antennenort s_{ij} fokussiert bleiben. Die Alternative von Bedingungsgleichungen anstelle einer Reparametrisierung von Sensorbeobachtungen am Sensorort s_{ij} durch die Bodyposition gilt neben GNSS auch für sonstige Sensoren. Die Verbesserungsgleichungen der weiteren o. g. MEMS- und MOEMS-Sensoren finden sich in [15] und [28].

3 Physikalische Grundlagen und Konzept der NAVKA-Flugsteuerung

Bei einem Multicopter mit $i=1,n$ Propellern (Abb. 1) ist die Flugdynamik einzig über die Propellerdrehzahlen ω_i bzw. deren fortlaufende Änderungen $\Delta\omega_i$ als Stellgrößen $\mathbf{u}(t)$ (1b) steuerbar.

Ausgehend von den propellerspezifischen Größen, des Schubbeiwerts c_i und des Propellerradius R_i , gilt für den Schubkraftvektor \mathbf{T}_i^b eines Propellers im Body-Koordinatensystem (b) eines UAV oder bemannten n-Propeller Multicopters die Gleichung (5a). Mit α_i bzw. δ_i werden die azimutale bzw. vertikale Orientierung der i -ten Propellerachse und mit \mathbf{r}_i der 3D-Positionsvektor des Propellers im Bodysystem (b) bezeichnet. Analog zur o. g. Behandlung verteilter Navigationssensoren erlaubt die NAVKA-Flugsteuerung auch ein freies Design zur räumlichen Verteilung und Orientierung der n-Propeller.

Schubkraft-vektor $\mathbf{T}_i^b = \rho \cdot \pi \cdot c_i \cdot R_i^4 \omega_i^2 \cdot (\cos \alpha_i \cdot \sin \delta_i, \sin \alpha_i \cdot \sin \delta_i, -\cos \delta_i)^T$ (5a)

Die entsprechenden fünf geometrischen Konfigurationsparameter pro Propeller sind $\mathbf{r}_i, \alpha_i, \delta_i$. Mit freien (α_i, δ_i) kann z. B. der Nachteil eines unteraktuierten Flugsystems bzw. Bodys (b) bei einheitlicher Propellerausrichtung in z-Richtung ($\delta_i=0^\circ$), wo nur vier der sechs Freiheitsgrade direkt regelbar sind, aufgehoben werden. Als freie physikalische Konfigurationsparameter treten propellerweise die o. g. Größen c_i und R_i hinzu. Die über die individuellen Propellerdrehzahlen ω_i als Elemente des Stellgrößenvektors $\mathbf{u}(t)$ (1b) erzeugten Schubkräfte \mathbf{T}_i^b (5a) und die direkt daraus resultierenden Propellerdrehmomente \mathbf{M}_i^b führen auf die folgenden Newton'schen bzw. Euler'schen Bewegungsgleichungen des Multicopter Bodys (b):

Newton'sche Bewegungsgleichungen $\mathbf{F}_{\text{ges}}^b = \sum_1^n \mathbf{T}_i^b + m \cdot \mathbf{R}_e^b(\mathbf{r}, p, y) \cdot \mathbf{g}^e + \mathbf{F}_{\text{env}}^b$ (5b)
 $= m \cdot \mathbf{R}_e^b \cdot \mathbf{R}_i^e \cdot \ddot{\mathbf{x}}^i$

Euler'sche Kreiselgleichungen $\mathbf{M}_{\text{ges}}^b = \sum_1^n \mathbf{M}_i^b + \mathbf{M}_{\text{env}}^b(d_i) = \boldsymbol{\omega}_{ib}^b \times (\mathbf{J} \cdot \boldsymbol{\omega}_{ib}^b) + \mathbf{J} \cdot \dot{\boldsymbol{\omega}}_{ib}^b$, (5c)
 mit $\mathbf{M}_i^b = \mathbf{r}_i \times \mathbf{T}_i^b$

Mit der Überführung der Beschleunigung $\ddot{\mathbf{x}}^i$ (5b) vom Inertialsystem (i) in das der Navigation mit $\mathbf{y}(t)$ (1) gegenständliche erdfeste System (e) (Abb. 1) treten infolge der Rotation \mathbf{R}_i^e zwischen inertialem (i) und erdfesten (e) System in (5b) noch die von den Navigationszustandsgrößen $\mathbf{y}(t)$ (1a) Geschwindigkeit ($\dot{x}^e \dot{y}^e \dot{z}^e$) und Position ($x^e y^e z^e$) abhängigen Coriolis- bzw. Fliehkraftterme hinzu. Links sind in (5b) – neben dem durch die gravitative Beschleunigung \mathbf{g}^e hervorgerufenen Kraftanteil – noch weitere luftwiderstands- und windinduzierte Umgebungs-(Environment)-Kräfte \mathbf{F}_{env}^b zu berücksichtigen. Mit m und \mathbf{J} werden die Masse und der Trägheitstensor des UAV bzw. des bemannten n-Multicopter Body (b) bezeichnet (Diskretisierung in Voxel, siehe Abb. 3).

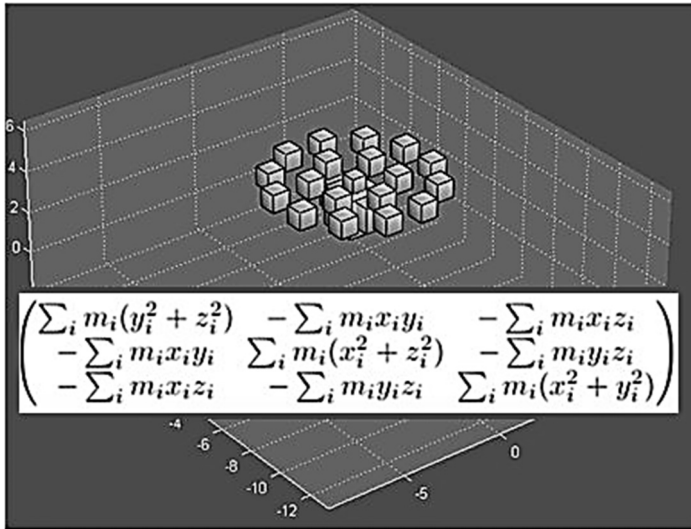


Abb. 3: Voxel-Diskretisierung des bemannten Volocopters und Berechnung des Trägheitstensors \mathbf{J} aus der Diskretisierung des Body (b) in Massenelemente

Auch bei den Drehmomenten (5c) kommen weitere – zum Teil vom Drehsinn d_i (+1 rechts-, -1 linksdrehend) abhängige – Propeller induzierte aerodynamisch bedingte Drehmomente $\mathbf{M}_{env}^b(d_i)$ – wie Roll-, Brems- und Horizontalkraft-Moment – hinzu. Die Bewegungsgleichungen (5b,c) implizieren ein bezüglich der sechs Freiheitsgrade – der drei 3D-Translationen ($x^e y^e z^e$) und der drei

3D-Rotationen ($\ddot{r}^e \ \ddot{p}^e \ \ddot{y}^e$) – inhärent instabiles gekoppeltes Differentialgleichungssystem. Schon allein deshalb ist der Navigationszustandsvektor $\mathbf{y}(t)$ (1) des Bodys (b) daher Gegenstand des Regelkreises der Flugsteuerung (Abb. 4) fortlaufend, standardmäßig in 400 bis 1.000 Hz, über die n Propellerdrehzahlen ω_i als Stellgrößenvektor $\mathbf{u}(t)$ (1b) zu regeln. Als Regeldifferenz, auch Regelabweichung $\mathbf{e}(t)$ tritt die Differenz zwischen dem mit der multisensoriellen Navigationskomponente der Flugsteuerung festgestellten Istzustand $\mathbf{y}(t)_{\text{Ist}}$ und dem – per Fernsteuerung, Piloten-Joystick oder Autopiloten – fortlaufend angeforderten Sollzustand $\mathbf{y}(t)_{\text{Soll}}$ auf. Es gilt:

$$\text{Allgemeine Regeldifferenz} \quad \mathbf{e}(t) = \mathbf{y}(t)_{\text{Soll}} - \mathbf{y}(t)_{\text{Ist}}. \quad (6)$$

Dabei beziehen sich die Soll- und Ist-Vorgaben i. A. nur auf einen Teil des vollständigen Navigationszustandsvektors $\mathbf{y}(t)$ (1a), z. B. die Drehraten und/oder die Orientierungswinkel Roll, Pitch, Yaw (Kap. 4 bis 6, Abb. 5).

Wie in [26] und [27] im Fall der Lageregelung (Abb. 5) dargestellt, können die Bewegungsgleichungen (5b,c) mit (6) wie folgt weiterentwickelt werden:

$$\begin{aligned} \text{NAVKA Flug-} \quad & \left[\mathbf{F}_{\text{ges}}^b, \mathbf{M}_{\text{ges}}^b \right]^T = \mathbf{F}(\mathbf{m}, \mathbf{J}, \mathbf{g}^c, \mathbf{y}(t), \dot{\mathbf{y}}(t)) \\ \text{steuerung.} \quad & = \mathbf{F}_{\text{PD}}(\mathbf{m}, \mathbf{J}, \mathbf{g}^c, \mathbf{y}(t), \mathbf{e}(t), \dot{\mathbf{e}}(t)) \end{aligned} \quad (7a)$$

$$\begin{aligned} \text{Stufe 1: Regeldiffe-} \quad & \\ \text{renz und physikali-} \quad & \\ \text{sches Modell} \quad & \left[\mathbf{F}_{\text{ges}}^b, \mathbf{M}_{\text{ges}}^b \right]^T =: \mathbf{u}'(t) = \mathbf{F}_{\text{PD}}(\mathbf{m}, \mathbf{J}, \mathbf{g}^c, \mathbf{e}(t), \dot{\mathbf{e}}(t)) \end{aligned} \quad (7b)$$

In den Gleichung (7a,b) treten die Komponenten der Regeldifferenzen $\mathbf{e}(t)$ und deren Ableitungen $\dot{\mathbf{e}}(t)$ mit insgesamt acht Faktoren $\mathbf{k}_{p,i}$ und $\mathbf{k}_{d,i}$ auf [26]. Es resultieren bei Änderungen $\mathbf{e}(t), \dot{\mathbf{e}}(t)$ im Ergebnis von (7b) eine neue Gesamtschubkraft und ein neues Drehmoment $\mathbf{u}'(t)$ (7b) zum Zeitpunkt t und damit in Stufe 1 entsprechende Änderungen $\Delta \mathbf{u}'(t)$ gegenüber der vorausgehend ermittelten Größe $\mathbf{u}'(t - \tau)$.

Mit (7a,b) lassen sich dann im Fall $n=4$ nach [26] und [27] auf analytischem Wege die Stellgröße neuer Drehzahlen $\mathbf{u}(t) = [\omega_1, \omega_2, \omega_3, \omega_4]^T$ bzw. die Drehzahländerungen $\Delta \omega_i$ berechnen.

Mit (7a,b) ist zugleich auch die Modellierung der Flugsteuerung von n-Propeller Multicoptern als sog. PD-Regler physikalisch begründet. Die Hinzunahme eines mit \mathbf{K}_I gehenden sog. integralen (I) Terms in (8) kompensiert die Regelabweichungen im Verlauf der Regeldauer τ . Wir erhalten die Stellgrößen in Stufe 1 alternativ zu (7a,b) damit als sog. PID-Regler:

Erweiterung von
NAVKA Stufe 1 auf
PID-Regelung

$$\mathbf{u}'(t) = \mathbf{K}_P \cdot \mathbf{e}(t) + \mathbf{K}_I \cdot \int_0^{\tau} \mathbf{e}(t) \cdot dt + \mathbf{K}_D \cdot \dot{\mathbf{e}}(t) \quad (8)$$

Die Diagonalmatrizen \mathbf{K}_P , \mathbf{K}_I und \mathbf{K}_D wirken proportional (P), auf den integrativen Teil (I) sowie auf die 1. Ableitung (D) der Regeldifferenz $\mathbf{e}(t)$, daher auch die Bezeichnung PID-Regler [25]. Die PID-Regelungsparameter \mathbf{K}_P , \mathbf{K}_I und \mathbf{K}_D sind, wie oben dargelegt, zum einen auf physikalischer Grundlage (Gleichungen (5a,b,c), (7a,b) – also mit den physikalischen Eigenschaften des zu regelnden Systems UAV teilweise oder komplett – ermittelbar ([26], [27]). Es ist jedoch darüber hinaus üblich, die Regelungsparameter experimentell z. B. mittels Flugsimulationsumgebungen oder adaptiv zu bestimmen.

In Stufe 2 wird aus (7a,b) bzw. der entsprechenden PID-Regelung (8), aus $\Delta \mathbf{u}'(t) = [\Delta \mathbf{F}, \Delta \mathbf{M}]^T$ in Verbindung mit (5a,b,c) der aktuelle Stellgrößenvektor, die Änderungen der Propellerdrehzahlen $\Delta \mathbf{u}(t) = [\Delta \omega_1, \dots, \Delta \omega_i, \dots, \Delta \omega_n]^T$ wie folgt erhalten.

NAVKA Flug-
steuerung

$$[\Delta \mathbf{F}, \Delta \mathbf{M}]^T = \mathbf{K} \cdot [\Delta \omega_1, \dots, \Delta \omega_i, \dots, \Delta \omega_n]^T \quad (9a)$$

Stufe 2: Drehzahl-
änderungen

$$\Delta \mathbf{u} = [\Delta \omega_1, \dots, \Delta \omega_i, \dots, \Delta \omega_n]^T = \mathbf{K}^{-1} \cdot [\Delta \mathbf{F}, \Delta \mathbf{M}]^T \quad (9b)$$

Der mit (8) und (9a,b) gegebene Ansatz zur Flugsteuerung mündet in (9b) auf die gesuchten finalen Stellgrößen der fortlaufenden Änderungen der n Propellerdrehzahlen $\Delta \mathbf{u}(t) = [\Delta \omega_1, \dots, \Delta \omega_i, \dots, \Delta \omega_n]^T$ in Funktion des Navigationszustands $\mathbf{y}(t)$ (1a) und dessen Änderung, der Regelabweichung $\mathbf{e}(t)$ (6).

Mit \mathbf{K}^{-1} (9b) wird jene i. A. wegen der infolge der Überbestimmtheit in (7c) nicht eindeutigen Stellgrößen $\Delta \omega_i$ verwendete generalisierte Inverse der Matrix \mathbf{K} (der sog. Motormatrix) bezeichnet, die zugleich auch die hinsichtlich der Beschränktheit der Drehzahländerungen zu richtenden Ungleichungen (12) erfüllt.

Der Ansatz (7a,b), (8), (9a,b) wurde im derzeitigen Entwicklungsstand der NAVKA Flight-Control NAVKArine-FC4 (Abb. 4) als kaskadierender PID Regler realisiert (Kap. 4). Die Kaskadierung in eine Positions- und eine Lage(Attitude)-Regelstrecke (Abb. 5) ergibt sich aufgrund der o. g. Unteraktuierung des Bodys (b) bzw. Flugsystems bei standardmäßigem Design einer durchgehend einheitlichen Propellerausrichtung in z-Richtung ($\delta_i=0^\circ$).

Abb. 4. zeigt die im NAVKA-Projekt am IAF/HSKA entwickelte Flight-Control Hardware NAVKArine-FC4 in Gestalt der geöffneten Box. Neben dem Mikrocontroller (ARM Cortex-M4) sind in der oberen Lage die MEMS InvenSense Sensorhardware, das GNSS-Board (u-blox L1 NEO M8T oder M8N) sowie das Funkmodul zu erkennen.

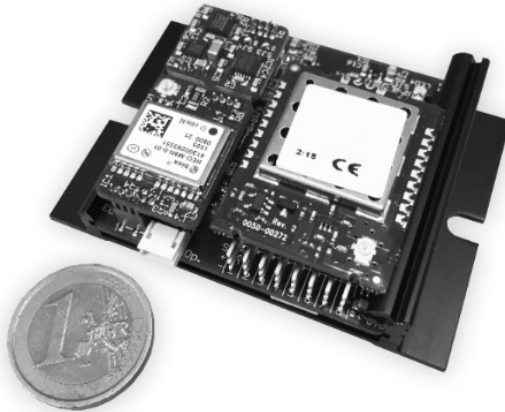


Abb. 4: GNSS/MEMS Navigations- und Flight-Control-Box NAVKArine-FC4

4 Realisierung von NAVKA-Flugsteuerung und Flugobjekten

4.1 PID-Einzelregler und weitere Regelungsverfahren

Die Abb. 5 zeigt die auf den Grundgleichungen zur PID-Regelung ((7a,b), (8), (9a,b)) basierende kaskadierende Regelung der NAVKA Flight-Control für den Abschnitt der Regelstrecke der Attitude(Lage)-Regelung, wie sie derzeit in Form

eines PID-Reglers (siehe auch [17]) für eine redundante Zahl von Navigations- und Flugsteuerungseinheiten („Flight-Control-Boxen“, Abb. 4) realisiert ist. Die Gleichungen für die PID-Regelung setzen an den physikalischen Grundgleichungen (5a,b,c) und (7a,b) an.

Neben der oben beschriebenen und in der derzeitigen NAVKA-Flight-Control NAVKArine-FC4 (Abb. 4, Abb. 7) als Steuerungskomponente implementierten PID-Regelung bestehen für die Regelung von Multicoptersystemen eine Reihe weiterer Ansätze. In [16] wird das Back-Stepping Regelungsverfahren auf die Regelung von einem Quadrocopter vorgestellt. Die Multicopterregelung im Zustandsraum (LQR) wird in [24] mit der PID Regelung verglichen. Adaptive Regelungsverfahren werden z. B. in [18], [19], [20] und [21] dargestellt und untersucht. Für UAVs wurden auch adaptive Regelungsansätze auf Basis von Neuronalen Netzen untersucht [22] (hier für Flächenflügler). Eine gute allgemeine Einführung in die Regelungstechnik von Multicoptern ist in [23] zu finden. Im Prinzip lassen sich aber alle genannten Ansätze in das in diesem Beitrag vorgestellte Framework integrieren.

Im Rahmen des Beitrags soll im Weiteren exemplarisch das Modell der PID Kaskadenregelung für die sog. Lageregelung erläutert werden. Hier bezieht sich die Vorgabe $y(t)_{\text{Soll}}$ auf drei Orientierungswinkel (r =Roll, p =Pitch, y =Yaw) als Bestandteile von (1a) (siehe, Abb. 5).

Die Lageregelung wird hier in einen Hauptregler (*äußerer* Regelkreis) und einen Hilfsregler (*innerer* Regelkreis) aufgeteilt. Dieser Aufbau hat den Vorteil, dass Störungen im Hilfsteil schon soweit ausgeglichen werden, dass sie im Hauptregler nur gering oder gar nicht bemerkt werden [25]. Durch diese Zerlegung in Teilprobleme lässt sich die Multicopterregelung in der Praxis leichter handhaben. Für den *inneren Regler* wird die Regeldifferenz $e(t)$ (6) bzgl. der Bodydrehraten gebildet, mit Hilfe derer die Soll-Orientierung hergestellt wird. Es gilt:

$$\begin{array}{ll} \text{Regeldifferenz} & \mathbf{e}(t) =: \Delta \boldsymbol{\omega}(t) = \boldsymbol{\omega}(t)_{\text{eb,Soll}}^b - \boldsymbol{\omega}(t)_{\text{eb,Ist}}^b \\ \text{innerer Regler} & \end{array} \quad (10a)$$

Die Regelabweichung $e(t)$ (10a) führt hier zu dem $\mathbf{u}'(t)$ (8) analogen entsprechenden Steuerungskommando \mathbf{f} . Für die i -te der drei Achsen (Roll, Pitch und Yaw) gilt:

$$\mathbf{f}_i(t) = \mathbf{K}_{p,i} \cdot \mathbf{e}_i(t) + \mathbf{K}_{I,i} \cdot \sum_0^{\tau} \mathbf{e}(t) dt + \mathbf{K}_{D,i} \cdot \frac{\mathbf{e}_i(t) - \mathbf{e}_i(t - \Delta t)}{\Delta t}. \quad (10b)$$

Pro Achse werden für den PID-Regler (10b) skalare Verstärkungswerte benötigt. $\mathbf{K}_{p,i}$ für den Proportionalanteil, $\mathbf{K}_{I,i}$ für den Integralanteil kumulierter Fehler $\Sigma \mathbf{e}$ der Regeldifferenz, sowie $\mathbf{K}_{D,i}$ für den D-Anteil. Prinzipiell sind $\mathbf{K}_{p,i}$, $\mathbf{K}_{I,i}$ und $\mathbf{K}_{D,i}$ pro Achse unterschiedlich; bei symmetrisch aufgebauten Multicoptern sind die Achsen Roll und Pitch meist mit identischen Werten zu betreiben. Das Steuerungskommando $\mathbf{u}'(t) =: \mathbf{f}(t)$ kann – wie in vorherigen Abschnitt mit (9a,b) beschrieben – mit Hilfe der generalisierten Inversen \mathbf{K}^- (9b) der Motormatrix auf die Motorstellgrößen der Drehzahländerungen für die einzelnen Motoren umgerechnet werden. Diese Umrechnung wird auch als *Motor Mixing* bezeichnet.

Für den *äußeren Regler* (Hauptregler) gilt wieder der Aufbau der PID-Reglergleichung und es gilt hier die Gleichung (10c).

$$\begin{aligned} \omega(t)_{\text{eb,Soll},i}^b = & \mathbf{K}_{\phi,p,i} \cdot \mathbf{e}_{\phi,i}(t) + \mathbf{K}_{\phi,I,i} \cdot \sum_0^{\tau} \mathbf{e}_{\phi,i}(t) dt \\ & + \mathbf{K}_{\phi,D,i} \cdot \frac{\mathbf{e}_{\phi,i}(t) - \mathbf{e}_{\phi,i}(t - \Delta t)}{\Delta t} \end{aligned} \quad (10c)$$

In (10c) werden die absoluten Winkelabweichungen in Roll, Pitch und Yaw als Regeldifferenzen $\mathbf{e}_{\phi,i}$ pro Achse mit eigenen PID-Faktoren $\mathbf{K}_{\phi,p,i}$, $\mathbf{K}_{\phi,I,i}$ und $\mathbf{K}_{\phi,D,i}$ eingeführt. Darüber werden die Solldrehzahlen im Body Frame (b) geregelt, die wiederum in den Regeldifferenzvektor $\omega(t)_{\text{eb,Soll}}^b$ (10a) einfließen.

Erst der innere Regelkreis (10a,b) errechnet dann aus $\mathbf{u}'(t) =: \mathbf{f}(t)$ (10b) fortlaufend die eigentlichen Motorstellgrößen $\Delta \mathbf{u} = [\Delta \omega_1, \dots, \Delta \omega_i, \dots, \Delta \omega_n]^T$ (1b) der Drehzahländerungen der n Propeller nach (9b). So entsteht im Ergebnis der beiden kaskadierenden Regelkreise mit den Bodydrehraten als gemeinsame Führungsgröße mit (10a,b,c) die Sollvorgabe $\mathbf{y}(t)_{\text{Soll}}$ bzgl. Roll, Pitch, Yaw (Abb. 5).

Es ergeben sich insgesamt 9 Parameter für den äußeren und nochmals 9 Parameter für den inneren Regelkreis der Lageregelung (Abb. 5). Für Euler-Winkel und Quaternionen ist der Aufbau vergleichbar – der Winkelabweichung ist einfach die Differenz zwischen dem gewünschten Winkel (Soll) der Lageregelung und dem aktuell aus der Navigationsschätzung $\mathbf{y}(t)$ (1a) bekannten Ist-Winkel.

Für die Bestimmung der insgesamt 18 Regelparameter ist es sinnvoll zuerst die des inneren Regelkreises zu bestimmen. Danach können die Parameter des äußeren Regelkreises ermittelt werden. Der innere Regelkreis ist an sich auch der kritische Regelkreis, der z. B. empfindlich auf Änderungen im Trägheitsmoment des Multicopters reagiert. In [21] ist z. B. nur der innere Regelkreis als adaptiver Regler implementiert.

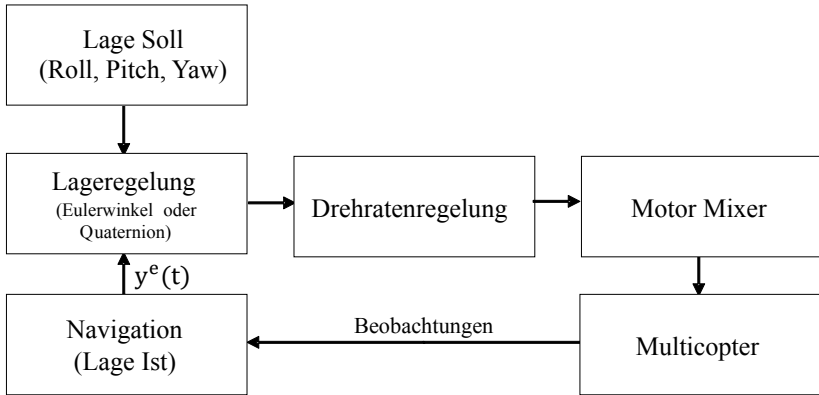


Abb. 5: Regelungsdiagramm für Plattform-redundante Flight-Control Boxen

4.2 Robuste Schätzungen von Navigationszustand und Einzelstellgrößen in Flugsteuerungssystemen

Mit Blick auf die o. g. mathematischen Modelle und Algorithmen für ein Flugsteuerungssystem für skalierbare UAV und bemannte n-Propeller Multicopter ergeben sich mehrere Positionen bzw. Stufen zur Herstellung eines fehlerrobusten Designs der Gesamtsteuerungssystems („Flight Control“) im Fall einer (Abb. 7) oder mehrerer – i. A. einer Anzahl m – physikalischer Flugsteuerungseinheiten (Abb. 9, Abb. 12) wie der NAVKarine-FC4 (Abb. 4). Dies sind:

- Grundsätzlich: Robuste Schätzung des Navigationszustands $\mathbf{y}(t)_{\text{Ist}}$ (1a) im Gauß-Markov-Modell von Zustandsvorhersage und GNSS/MEMS/MOEMS-Sensordaten (2a,b). Damit ist zugleich auch die steuerungsrelevante Regeldifferenz $\mathbf{e}(t) = \mathbf{y}(t)_{\text{Soll}} - \mathbf{y}(t)_{\text{Ist}}$ (6) robust.
- Robuste Schätzung von $\mathbf{y}(t)_{\text{Ist}}$ und damit $\mathbf{e}(t)$ unter Zusammenführung der Sensordaten einer Anzahl m physikalisch getrennter Flugsteuerungseinheiten

zu einem virtuellen Sensornetz und Übergabe von $\mathbf{e}(t)$ an die Steuerungsalgorithmen zur Bestimmung von $\mathbf{u}(t)$ (1b) nach (9b) bzw. (10a,b,c). Dies entspricht der jetzigen Realisierung der Ermittlung eines gemeinsamen Navigationszustands- und Stellgrößenvektors $\mathbf{y}(t)$ (1a) bzw. $\mathbf{u}(t)$ (1b) der NAVKA-Flugsteuerung FC4.

- Getrennte robuste Schätzung einzelner $\mathbf{y}(t)_{\text{ist}}-i$ und $\mathbf{e}(t)-i$ in den physikalisch getrennten Flugsteuerungseinheiten. Ermittlung einzelner (möglicherweise fehlerhafter) Stellgrößen $\mathbf{u}(t)-i$. Erneute robuste Gesamtschätzung eines finalen $\mathbf{u}(t)$ Stellgrößenvektors der Propellerdrehzahlen. Die physikalische Trennung sorgt insbesondere hinsichtlich eines bemannten Flugs für erhöhte technische Sicherheit. Dieses aktuell in Entwicklung befindliche NAVKA-Flugsteuerungskonzept (FC5) zur Ermittlung eines gemeinsamen Navigationszustands- und Stellgrößenvektors ist in Abb. 6 und Abb. 9 dargestellt. Es wird in den Kap. 5 und 6 vorgestellt.

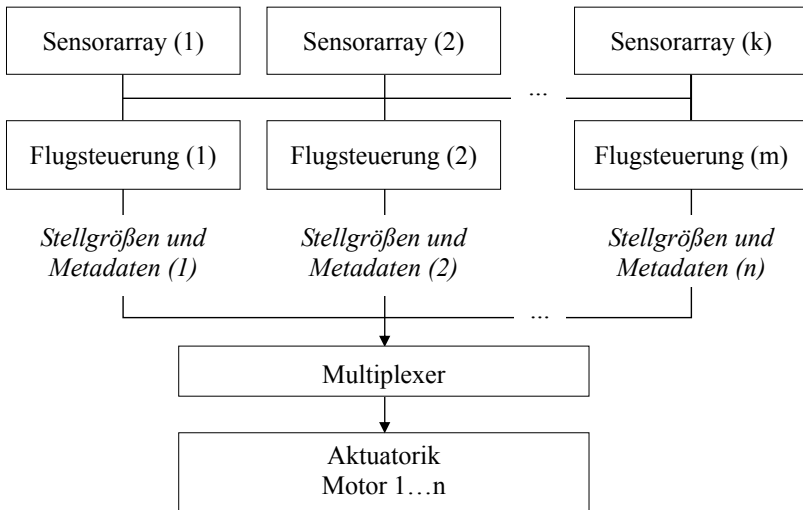


Abb. 6: NAVKA Flight-Control.

Regelstrecke der Lage-(Attitude) Steuerung im kaskadierenden PID-Regler

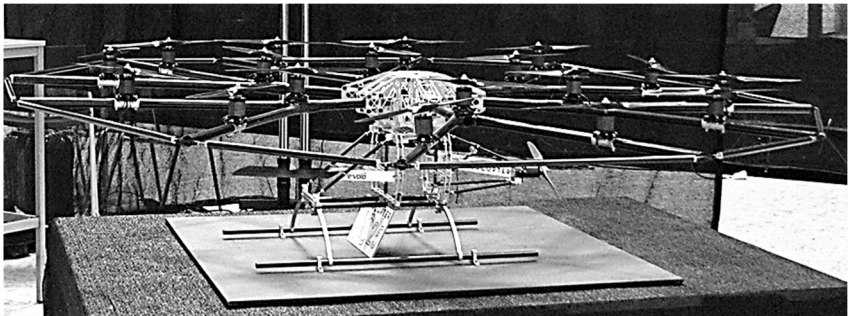
Die Abb. 7 zeigt die auf dem NAVKA Klein-UAV ($n=6$ Propeller) verbaute NAVKArine-FC4 mit Funkantenne und GNSS-Antennenkabel an den entsprechenden Schnittstellen-Ausgängen. Hiernach können die entwickelten Flight-

Control-Boxen (Abb. 4) auch lediglich als Navigationssensor-redundante Einzelplattformen zum Einsatz kommen.

Die Abb. 8. zeigt das Test-UAV VC 25 ($n=18$ Propeller, Durchmesser 3 m, Tragkraft bis 50 kg).



*Abb. 7: NAVKA Klein-UAV mit NAVKArine-FC4 Box.
Videos zu Testflügen finden sich auf www.navka.de/index.php/de/videos.*



*Abb. 8: UAV VC 25 mit Karbonrahmen
(25 kg, 3 m, $n=18$ Propeller, Tragkraft ca. 50 kg)*



*Abb. 9: Bemannter Multicopter VC 200
(Payload 450 kg, n=18 Propeller)*

Die Abb. 9 den als bemanntes Flugobjekt VC 200 in ZIM Verbundforschung (ZIM VP2957702TL2) bei sechs Konsortiumpartnern realisierten Multicopter VC200 (n=18 Propeller, Tragkraft 450 kg). Bei diesem ZIM-Projekt wurde NAVKA-seits ebenfalls die zugehörige Navigations- und Steuerungskomponente als Plattform-redundante Flight-Control nach Abb. 6 entwickelt.

5 L₁-basierte robuste Auswahllogik für die Aktorik

Die Grundidee dieses auf einer Anzahl von m einzelnen Steuerungsboxen basierenden robusten Gesamtsystems basiert auf der Idee, die Auswahl der Motorstellgrößen zu jedem Zeitpunkt t als Schätzproblem zu formulieren:

$$\mathbf{l}(t) + \mathbf{v} = \mathbf{A} \cdot \mathbf{x}(t) \quad (11)$$

Der Vektor \mathbf{x} sei der gesuchte bzgl. $\mathbf{l}(t)$ fehlerrobust geschätzte Vektor mit Motorstellgrößen. Der Vektor \mathbf{l} enthält entsprechend jeweils die von den verschiedenen Flugsteuerungen berechnete Lösung als zusammengesetzten Spaltenvek-

tor $\mathbf{l} = (\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_m)^T$ als funktionales Modell bzw. Beobachtungsgleichungen des Gauß-Markov-Modells (11). Die Designmatrix \mathbf{A} ordnet die Einzellösungen der Gesamtlösung \mathbf{x} zu. Die Motorstellgrößen können dabei entweder direkt für jeden Motor einzeln als Prozentangaben ausgedrückt werden oder alternativ als Roll-, Pitch-, Yaw- und Schubkommando zusammengefasst werden (um dann in einem zweiten Schritt in Motorstellgrößen umgerechnet zu werden, wie in Abschnitt 2 beschrieben). Im Folgenden soll mit den resultierenden Motorstellgrößen gearbeitet werden. Dadurch hat der Vektor \mathbf{x} als Länge die Anzahl der Motoren n , der Vektor \mathbf{l} hat die Länge m (Anzahl Flugsteuerungen) $\times n$ (Anzahl Aktuatoren). Die Matrix \mathbf{A} besteht aus gestapelten Einheitsmatrizen mit der Gesamtdimension: $(m \cdot n) \times (n)$.

Gleichung (11) könnte als Ausgleichungsproblem nach der Methode der kleinsten Quadrate (L_2 -Norm) gelöst werden, dadurch wäre jedoch der Einfluss von Fehlern (in einer oder mehreren Flugsteuerungen) nicht begrenzt (Einflussfunktion $|\Psi(\mathbf{v})|$ muss beschränkt sein, siehe [7]). Dies ist bei der L_1 -Norm gegeben.

Denkbar wären theoretisch auch andere robuste Schätzer. Der Vorteil der L_1 -Norm ist, abgesehen von der Robustheit, dass sich eine Lösung mit den Methoden der linearen Programmierung bzw. der linearen Optimierung auf Basis von modifizierten Simplex-Algorithmen finden lässt. Dadurch eröffnet sich die Möglichkeit auch Ungleichungen in folgender Form einzubeziehen:

$$\mathbf{C} \cdot \mathbf{x}(t) \leq \mathbf{d}(t). \quad (12)$$

Durch eine Multiplikation mit -1 können abweichende Vorgaben wieder in die Form von Gleichung (12) gebracht werden.

Das Ausgangsproblem ist das robuste Schätzkonzept [7] der Minimierung der Summe der absoluten Verbesserungen $|\mathbf{v}|$ aus Gleichung (11), sowie die Einhaltung der Bedingungen von Gleichung (12). Auf die vorausgehende Homogenisierung [7] des GMM (11) kann verzichtet werden, da die Beobachtungen $\mathbf{l}(t)$ (im jetzigen Stand der Entwicklungen) in (11) als unkorreliert und gleich genau betrachtet werden. Zudem kann eine Homogenisierung auf der Grundlage eines falschen stochastischen Modells selbst wieder eine Fehlerquelle darstellen. Zur Minimierung der L_1 -Norm wird die auf Georg Dantzig (1947) zurückgehende Simplex-Methode verwendet. Diese löst die Aufgabenstellung durch das Gauß-Jordan Eliminationsverfahren. Verschiedene Autoren haben den ursprünglichen Algorithmus dahingehend erweitert, dass die benötigte Rechenzeit verkürzt wird und der Speicherverbrauch minimiert wird. Eine gute Übersicht sowie ein Ver-

gleich der Algorithmen ist in [9] zu finden. Von der Geschwindigkeit her sind laut [8] und [9] besonders die Ansätze von [10], [11] und [12] hervorzuheben. Hervorzuheben ist noch der Beitrag von [13], hier wird ein modifizierter Simplex-Algorithmus vorgestellt, der effizient eine L_1 -Norm Lösung berechnet und sowohl Ungleichungen (12) als auch streng einzuhaltende Bedingungsgleichungen in folgender Form berücksichtigen kann:

$$\mathbf{E} \cdot \mathbf{x}(t) = \mathbf{f}(t).$$

In Abb. 10 ist ein einfaches Beispiel für einen Quadrokopter aufgestellt, bei dem drei Flugsteuerungen eine mögliche Motoransteuerung zu einem Zeitpunkt t zur Verfügung stellen. Gesucht ist der Vektor $\mathbf{x}(t)$ zur tatsächlichen Ansteuerung der Motoren. Der Übersichtlichkeit halber sind die Motorstellgrößen in Prozent (%) angegeben. Numerisch kann in einer Implementierung auch mit Werten zwischen 0 und 1 gearbeitet werden. Während sich in diesem Beispiel die Flugsteuerungen 2 und 3 in etwa bestätigen, liegt ein Problem bei Flugsteuerung 1 vor.

$\mathbf{l}(t)$ (%)		\mathbf{A}		$\mathbf{x}(t)$ (%)
256		1 0 0 0	·	X1
256		0 1 0 0		X2
256		0 0 1 0		X3
256		0 0 0 1		X4
56	=	1 0 0 0		
54		0 1 0 0		
46		0 0 1 0		
44		0 0 0 1		
55		1 0 0 0		
55		0 1 0 0		
45		0 0 1 0		
45		0 0 0 1		

Abb. 10: Beispiel für einen Multicopter mit vier Motoren

Die Werte widersprechen nicht nur den beiden anderen Lösungen, sie übersteigen auch den zulässigen Bereich (zwischen 0 % und 100 %). Die Ungleichungen sind in Abb. 11 zu sehen. Der Minimalschub ist hier beispielhaft mit 10 % angegeben, da ein Abschalten der Motoren im Flug ebenfalls zu vermeiden ist.

Der Aufbau von Abb. 6 enthält mit dem Multiplexer-Element einen sogenannten „Single Point of Failure“, was dem Prinzip der Redundanz offensichtlich widerspricht. Daher soll die Hardwareumgebung so erweitert werden, dass an jedem Motor die Multiplexerlogik bzw. die Auswahllogik vorhanden ist. Dieser Aufbau ist in Abb. 12 zu sehen. Wichtig ist hier, dass die Anbindung der Aktorik ebenfalls ausfallsicher erfolgt. Z. B. durch Mehrfachanbindung. Der Aufwand und damit auch die Kosten sind bei diesem Ansatz deutlich höher, sodass der Aufbau in Abb. 6 als Kompromiss verstanden werden kann, falls aufgrund von Platzmangel oder Kostengründen keine volle Redundanz möglich ist, aber dennoch eine erhöhte Sicherheit gewünscht ist. Mit üblichen Motorcontrollern (Electronic Speed Controller) beispielsweise ist dieses Konzept nicht umzusetzen.

C

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

-1	0	0	0
0	-1	0	0
0	0	-1	0
0	0	0	-1

x(t)

X1
X2
X3
X4

d(t)

100
100
100
100

-10
-10
-10
-10

\cdot

\leq

Abb. 11: Ungleichungen für vier Motoren

6 Implementierung und Simulation

Für die Synchronisierung der Motorstellgrößen muss ein Zeitfenster vorgegeben werden: ΔT . Je nach gewünschter Frequenz der Multicopterregelung leiten sich Grenzen im Millisekundenbereich ab. Typische Werte liegen zwischen 1 bis 10 ms. Nicht etwa, weil die Motoren so schnell auf neue Stellgrößen reagieren können, sondern um Verzögerungen möglichst gering zu halten. Jede Flugsteuerung muss in diesem Bereich eine mögliche Lösung zur Verfügung stellen.

Für die konkrete Implementierung muss immer eine Flugsteuerung als aktive Lösung gewählt werden, sodass die Flugsteuerung, die mit minimaler Verbesserungsbetragssumme an **x** heranreicht als aktive Flugsteuerung gewählt wird.

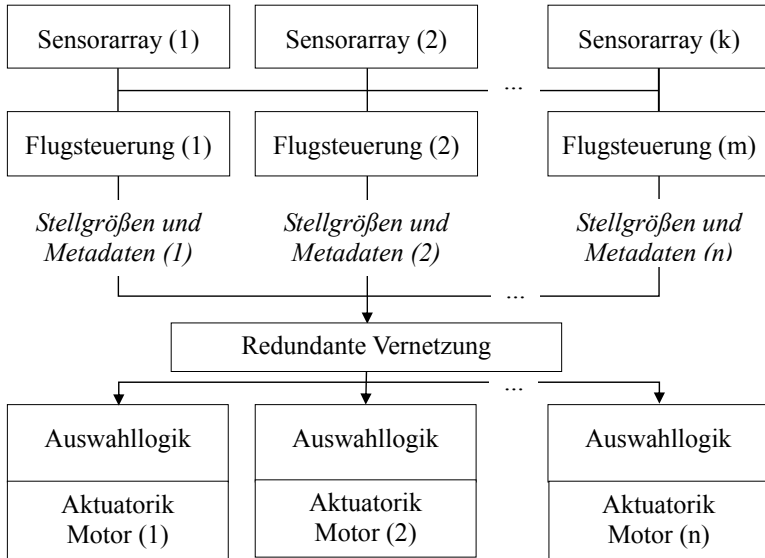


Abb. 12: Verteilung der Auswahllogik auf die Aktuatorik

D. h. nicht die Lösung \mathbf{x} wird an die Motoren ausgegeben, sondern die Flugsteuerung die nach dem genannten Kriterium an \mathbf{x} heranreicht. Die aktive Flugsteuerung hat dann einen Vorteil gegenüber den übrigen Flugsteuerungen, da sie sich im *Closed-Loop* Modus befindet, also aktiv wirken kann. Die übrigen Flugsteuerungen befinden sich im *Open-Loop* Modus. Eine adaptive Anpassung der Regelgrößen ist für die Flugsteuerungen im *Open-Loop* Modus nur begrenzt sinnvoll, da die ausgegebenen Stellgrößen nicht direkt auf das Flugobjekt einwirken. Daher hat es sich in den simulativen Tests als günstig erwiesen, zwischen adaptiven Motorstellgrößen und deterministischen Motorstellgrößen zu trennen und nur die deterministischen Motorstellgrößen in das vorgestellte Schätzkonzept zu integrieren. Die aktive Flugsteuerung gibt die Summe der beiden Stellgrößen aus:

$$\mathbf{u}_m(t) = \mathbf{u}_{\text{det}}(t) + \mathbf{u}_{\text{adapt}}(t) \quad (14)$$

Der deterministische Teil errechnet sich dabei allein aus dem aktuellen Navigationszustandsvektor. Der adaptive Teil kann durch unterschiedliche Regelungskonzepte bestimmt werden. Ein einfaches Beispiel wäre der in Abschnitt 4 vorgestellte Integralanteil eines PID-Reglers.

Zur Validierung des Konzepts wurden die Algorithmen in der NAVKA Multi-copter Simulationsumgebung so eingebunden, dass die selben C/C++ Routinen, die auf der Microcontrollerebene laufen, als MATLAB MEX Module in die Simulationsumgebung eingebunden werden können. Dazu wurde der Algorithmus von [13] als C++ Code auf Basis der *Eigen*-Mathematikbibliothek umgesetzt und ist damit das Herzstück der robusten Schätzung mit Messungen und Ungleichungen. Ein Hexakopter wurde im Geradeausflug (in Richtung der Rollachse) mit einem Pitchwinkel von -35° geflogen (Abb. 13). Dabei wurde in Sekunde 61,137 (Simulationszeit) die aktive Flugsteuerungslösung mit Zufallszahlen überschrieben um einen Ausfall zu simulieren. Die Flugsteuerungen 2 und 3 (Gesamtanzahl an Flugsteuerungen: 3) liefen weiter. Die Auswirkungen sind zumindest im Höhenregler (Abb. 14) zu erkennen, hier verliert der Multicopter kurzzeitig zwischen Sek. 61,137 und 64 etwa 7 cm an Höhe. Das ist auf den adaptiven Teil im Höhenregler der neu-aktivierten Flugsteuerung (hier #2) zurückzuführen.

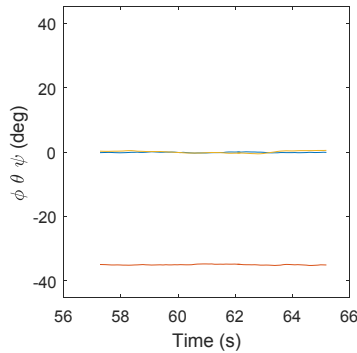


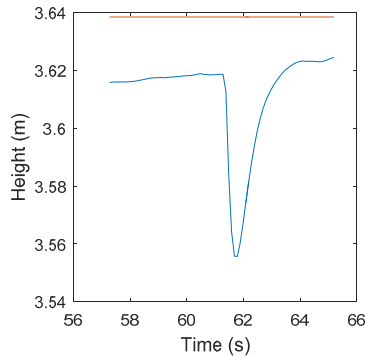
Abb. 13: Orientierung (Roll, Pitch, Yaw) während Ausfall von Flugsteuerung #1 (Roll/Yaw bei 0° , Pitch bei -35°)

7 Zusammenfassung und Ausblick

Im Rahmen des IAF Projektes NAVKA wurde am Institut für Angewandte Forschung (IAF) der Hochschule Karlsruhe ein allgemeines Flight-Control-System (Hardware, Algorithmen und Software) für – aus Sicherheitsgründen insbesondere im bemannten Flug – redundant verteilte GNSS/MEMS Navigationssensoren und -Plattformen realisiert. Redundanz sowie freie Konfigurierbarkeit beste-

hen für die NAVKA Flight-Control auch in Bezug auf n Propeller. Damit leistet die NAVKA Flight-Control neben einer so möglichen Aktuiertheit des Flugsystems auch die Skalierbarkeit von Multicopter-Flugsystemen, welche mit der NAVKA Flight-Control per Remote Control, per Piloten-Joystick (bemannt) oder im autonomen Flug (Auto-Pilot) betrieben werden können.

Skalierbarkeit besteht in Bezug auf Größe, Einsatzbereich und Systemdesign. Neben bemanntem Flug bestehen so in vielen Gebieten Entwicklungspotenziale für UAV-Systeme (UAS). Vorneweg zu nennen sind 3D out-/indoor Mapping-UAS mit Kameras und Laserscannern. Entwicklungen dazu erfolgen aktuell auch im NAVKA-Projekt [6]. Weitere Beispiele sind ABC-Geosensing UAS, Inspektions-UAS für Bauwerke, UAS für out-/indoor-Facility Management und Observierung von Produktionsabläufen, UAS zur Personenbergung bei diversen SAR (Save & Rescue) Szenarien, Feuerbekämpfungs-UAS, Transport-UAS, UAS für die Filmbranche, UAS in Land- und Forstwirtschaft, u. v. a.



*Abb. 14: Höhe von Hexakopter in Simulation.
Ausfall bei Sekunde 61,137*

Die gegenwärtigen FuE zur NAVKA Flight-Control sind mit Aufdeckung und Regelung von Änderungen in Masse und Trägheitstensor von UAS und bemannten Multicoptern befasst. Darüber hinaus steht die Erweiterung der Navigationskomponente um optische Sensoren (MOEMS, Kameras, Laserscanner) im Fokus. Damit wird die Flugfähigkeit von UAS sowie bemannten Systemen in GNSS abgeschatteten Bereichen verbessert. UAS können damit automatisierte Anflug- und Landemanöver auf Objekten (z. B. bewegten Fahrzeugen für SAR) durchführen, und schließlich wird so z. B. in FuE wie u. a. [6] deren Indoorfähigkeit erschlossen.

Das vorgestellte Redundanzkonzept auf Basis der L_1 -Norm für Multicopter erlaubt einen sicheren und robusten Betrieb von mehrfach installierten Flugsteuerungen inkl. verteilter Navigationssensoren.

Literatur

- [1] ESNC 2015 Website: www.esnc.info/index.php?anzeige=bawue14.html.
- [2] Webseite NAVKA Projekt: www.navka.de.
- [3] Webseite ZIM Projekt Volokopter: www.navka.de/index.php/de/ueberblick-evolo.
- [4] Jäger, R. (2014): „Navigation mit verteilten GNSS/MEMS Sensoren – Mathematische Modelle, Algorithmen und Anwendungspotenziale“. Horizonte (43). ISSN 1432-0174. S. 7–10.
- [5] Zwiener, J.; R. Jäger (2015): Flight-Control and Navigation for Scalable and Arbitrarily Dimensioned UAV and Manned Multicopters. Proceedings of Interexpo-GeoSiberia 2015, April 2015. Siberian State Academy of Geodesy (SSGA). ISBN 978-5-87693-803-9. S. 98-108.
- [6] Webseite Innovative-Projekte Projekt MSM: www.navka.de/index.php/de/ueberblick-msm.
- [7] Jäger, R.; Müller, T.; Saler, H.; Schwäble, R. (2005). Klassische und robuste Ausgleichungsverfahren. Wichmann, Heidelberg.
- [8] Abdelmalek, N.; Malek, W. A. (2008). Numerical linear approximation in C. CRC Press.
- [9] Späth, H. (1987). Mathematische Software zur linearen Regression Oldenbourg Verlag. ISBN: 3-486-20375-4.
- [10] Abdelmalek, N. N. (1980). L_1 solution of overdetermined systems of linear equations. ACM Transactions on Mathematical Software (TOMS), 6(2), 220–227.
- [11] Armstrong, R. D.; Frome, E. L.; Kung, D. S. (1979). A revised simplex algorithm for the absolute deviation curve fitting problem. Communications in Statistics-Simulation and Computation, 8(2), 175–190.
- [12] Barrodale, I.; Roberts, F. D. (1973). An improved algorithm for discrete L_1 linear approximation. SIAM Journal on Numerical Analysis, 10(5), 839–848.
- [13] Barrodale, I.; Roberts, F. D. K. (1978). An Efficient Algorithm for Discrete L_1 Linear Approximation with Linear Constraints. SIAM Journal on Numerical Analysis, 15(3), 603–611.
- [14] Jäger, R.; Gonzalez, F. (2005): GNSS/GPS/LPS based Online Control and Alarm System (GOCA) – Mathematical Models and Technical Realisation

- of a System for Natural and Geotechnical Deformation Monitoring and Hazard Prevention. ISGDM IAG-Symposium 2005. University of Escuela Politécnica Superior de Jaén. Spain. (Sanso, F.; Gil, A. J. (Eds.): Geodetic Deformation Monitoring: From Geophysical to Engineering Roles. IAG Series on Geodesy Symposia. Springer Heidelberg and New York. ISBN 3-540-38595-9. S. 293–304.
- [15] Jäger, R. (2015): Reiner Jäger: NAVKA – Navigation Algorithms and Technologies for Distributed GNSS/MEMS/MOEMS Sensors. Vortrag Challenge Workshop Parameter and State Estimation Methods – Software – Applications, 10.–11. November 2015, IWR (Interdisciplinary Center for Scientific Computing 2015), Heidelberg University. Download at: www.navka.de/images/files/JÄGER_HCO_11_2015_INTERNET.pdf.
 - [16] Meister, O. (2010): Entwurf und Realisierung einer Aufklärungsplattform auf Basis eines unbemannten Minihelikopters mit autonomen Flugfähigkeiten. Logos Verlag Berlin GmbH.
 - [17] Puls, T. (2011): Lokalisations- und Regelungsverfahren für einen 4-Rotor-Helikopter. Dissertation, Universität Oldenburg.
 - [18] Dydek, Z. T. (2010): Adaptive control of unmanned aerial systems. Dissertation. Massachusetts Institute of Technology.
 - [19] Dydek, Z. T.; Annaswamy, A. M.; Lavretsky, E. (2013): Adaptive control of quadrotor UAVs: A design trade study with flight evaluations. *Control Systems Technology, IEEE Transactions on*, 21(4), 1400–1406.
 - [20] Zhou, W.; Yin, K.; Wang, R.; Wang, Y. E. (2014): Design of Attitude Control System for UAV Based on Feedback Linearization and Adaptive Control. *Mathematical Problems in Engineering*, 2014.
 - [21] Achteik, M.; Bierling, T.; Wang, J.; Hocht, L.; Holzapfel, F. (2011): Adaptive Control of a Quadcopter in the Presence of large/complete Parameter Uncertainties. *Infotech@ Aerospace 2011*, 2011–1485.
 - [22] Schumacher, C. J.; Kumar, R. (2000): Adaptive control of UAVs in close-coupled formation flight. In *American Control Conference*, 2000. *Proceedings of the 2000* (Vol. 2, pp. 849–853). IEEE.
 - [23] Mahony, R.; Kumar, V.; Corke, P. (2012): Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *IEEE Robotics & Automation Magazine*, (19), 20–32.
 - [24] Argentim, L. M.; Rezende, W. C.; Santos, P. E.; Aguiar, R. (2013): PID, LQR and LQR-PID on a quadcopter platform. In *Informatics, Electronics & Vision (ICIEV)*, 2013 International Conference on (pp. 1–6). IEEE.
 - [25] Unbehauen, H. (1989): *Regelungstechnik*. Braunschweig: Vieweg.

- [26] Teppo Luukkonen (2011): Modelling and Control of Quadcopter School of Science. Mat-2.4108. Independent Research Project in Applied Mathematics. Espoo. Schweden.
- [27] Dikmen, I. C.; Arısoy, A.; Temelta, H. (2009): Attitude Control of a Quadrotor. Proceedings 4. International Conference on Recent Advances in Space Technologies, Istanbul. pp. 722–727.
- [28] Jäger, R. (2015): NAVKA – Aktuelle Systementwicklungen zur multisensoriellen Indoornavigation und zur Erfassung virtueller Gebäude. Vortrag Fachaustausch Geoinformation, 25. November 2015, Heidelberg, Print Media Center, Veranstalter Netzwerk GeoNet.MRN. Download at: www.navka.de/images/files/JÄGER_GEONETMRN_11-2015_INTERNET.pdf.